

DEVELOPMENT OF IMAGE PROCESSING SOFTWARE  
USING VISUAL BASIC 6.0

NURUL HAKEEM BINTI ABD RAHIM

UNIVERSITI MALAYSIA PAHANG

“I hereby acknowledge that the scope and quality of this thesis is qualified for the  
award of the Bachelor Degree of Electrical Engineering (Hons.) (Electronics)”

Signature : \_\_\_\_\_

Name : NURULFADZILAH BINTI HASSAN

Date : 16 MEI 2008

DEVELOPMENT OF IMAGE PROCESSING SOFTWARE USING  
VISUAL BASIC 6.0

NURUL HAKEEM BINTI ABD RAHIM

This thesis is submitted as partial fulfillment of the requirements for the award of the  
Bachelor Degree of Electrical Engineering (Hons.) (Electronics)

Faculty of Electrical & Electronics Engineering  
Universiti Malaysia Pahang

MEI 2008

“All the trademark and copyrights use herein are property of their respective owner. References of information from other sources are quoted accordingly; otherwise the information presented in this report is solely work of the author.”

Signature : \_\_\_\_\_

Author : NURUL HAKEEM BINTI ABD RAHIM

Date : 16 MEI 2008

## **ABSTRACT**

Image processing is any form of information processing when both the input and output are images. The process does not reduce the amount of data present but rearranges it. Therefore, the development of image processing software is to enable image processing functions to be applied to input images to acquire an output images that has been processed using this software. The development of this software was done using Visual Basic 6.0 and includes basic image processing functions such as blurring and sharpening, edge detection and colour balancing. This software was designed to be user-friendly with a simple interface, yet able to perform basic image processing functions effectively. The input images also can be imported from attached webcam. Overall, this development of image processing software is to ensure image processing to be done using less memory space approximately 380KB compared to existing image processing software which may reach up to 1GB.

## **ABSTRAK**

Pemprosesan imej ialah sebarang jenis proses imej apabila kedua-dua masukan dan keluaran ialah imej. Proses ini tidak mengurangkan jumlah keseluruhan data yang ada tetapi hanya menyusun semula data tersebut dalam bentuk yang berbeza. Pembangunan perisian ini adalah untuk membolehkan fungsi imej diaplikasikan kepada imej masukan untuk mendapatkan imej keluaran yang telah diproses menggunakan perisian ini. Pembangunan perisian imej ini dijalankan menggunakan Visual Basic 6.0. Ianya termasuklah fungsi asas pemprosesan imej seperti melaraskan ketajaman imej, pengesanan sempadan dan imbalan warna. Perisian ini dirancang untuk menjadi perisian yang mesra pengguna dengan aplikasi yang ringkas, namun berkebolehan untuk menjalankan fungsi asas pemprosesan imej dengan berkesan. Imej masukan, selain boleh didapati dari imej yang sedia ada, boleh juga didapati terus dari kamera web . Secara keseluruhannya, pembangunan perisian imej ini adalah untuk memastikan pemprosesan imej dijalankan dengan berkesan dengan hanya ruang memori yang kecil, iaitu 380KB berbanding dengan perisian imej lain yang sedia ada yang kadangkala boleh mencecah saiz sehingga 1GB.

## TABLE OF CONTENTS

CHAPTER	TITLE	PAGE
	TITLE PAGE	i
	DECLARATION	ii
	ABSTRACT	iii
	ABSTRAK	iv
	TABLE OF CONTENTS	v
	LIST OF FIGURES	viii
	LIST OF TABLES	x
	LIST OF APPENDICES	xi
 <b>1</b>	 <b>INTRODUCTION</b>	
	1.1 History of image processing	1
	1.2 Problem statement	2
	1.3 Objective	2
	1.4 Scope of project	3

## 2 LITERATURE REVIEW

2.1	Definition of image processing	4
2.2	Image processing functions/algorithms	5
2.2.1	Spatial filtering	6
2.2.1.1	Sharpening	6
2.2.1.2	Blurring	7
2.2.1.3	Edge detection	8
2.2.2	Diffuse	9
2.3	Introduction to Visual Basic 6.0	9
2.4	The waterfall model	10
2.5	Application of image processing	13

## 3 METHODOLOGY

3.1	Implementation of waterfall model	14
3.1.1	Planning	14
3.1.2	Requirement analysis	15
3.1.3	System design	15
3.1.3.1	Software interface	18
3.1.4	Implementation	24
3.1.5	Integration and testing	29
3.2	Implementation of basic image processing functions	31
3.2.1	Sharpen and blurring (High pass and low pass filters)	31
3.2.2	Grey scale	34
3.2.3	Colours	35
3.2.4	Edge detection	36



**4                      RESULT AND DISCUSSION**

4.1	Result	38
4.1.1	Software start up	38
4.1.2	Basic image processing function	42
4.1.3	Capturing from webcam	47
4.1.4	Other functions	49
4.2	Discussion	52

**5                      CONCLUSION AND RECOMMENDATION**

5.1	Conclusion	53
5.2	Recommendation for future work	54
5.3	Costing and commercialization	54

<b>REFERENCES</b>	<b>55</b>
-------------------	-----------

<b>APPENDICES</b>	<b>57</b>
-------------------	-----------

## LIST OF FIGURES

<b>FIGURE NO</b>	<b>TITLE</b>	<b>PAGE</b>
2.1	The waterfall model	11
3.1	The software operation flowchart	17
3.2	Introduction window	18
3.3	Main page interface	19
3.4	Webcam window	20
3.5	A control to determine image size in pixels	21
3.6	Filter browser interface	22
3.7	Image properties	23
3.8	Resize image interface	23
3.9	Introduction interface	24
3.10	Webcam browser programmed	25
3.11	Filter browser interface	27
3.12	Full compile of project	29
3.13	Error in compiling	30
3.14	Shortcut icon of the software	30
4.1	Start up of the program	39

4.2	User being asked to determine the size of window to be opened	39
4.3	Checkbox to determine type of image that can be handled	40
4.4	Open image	41
4.5	Image selected will be displayed	41
4.6	Dropdown “Image” menu show options of functions that are available to be applied to image	43
4.7	Filter browser	44
4.8	Load webcam	47
4.9	Webcam window	48
4.10	Image captured will be displayed after “Load image” button clicked	48
4.11	Resize image options	49
4.12	Zoom in and out options	50
4.13	Print options	50
4.14	Save dialog box that appears when user tries to close the window before saving the work	51

**LIST OF TABLES**

<b>TABLE NO</b>	<b>TITLE</b>	<b>PAGE</b>
3.1	Modules and their purposes	28
4.1	The commercial name used for image processing functions	42
4.2	Original image and after effect results	44

**LIST OF APPENDICES**

<b>APPENDIX</b>	<b>TITLE</b>	<b>PAGE</b>
A	Gantt Chart for PSM1 & PSM2	57
B	Visual Basic source code (module)	59
C	Visual Basic source code (interface)	78

## **CHAPTER 1**

### **INTRODUCTION**

#### **1.1 History of Image Processing**

Image processing or widely called digital image processing were developed in 1960s and among the earliest developers are Jet Propulsion Laboratory, MIT and Bell Labs. It was originally developed with application to satellite imagery, wire photo standards conversion, medical imaging, videophone, and character recognition and photo enhancement. Since the cost of computing back then was too high, it was only until 1970s that the area began to flourish with cheaper computers and available dedicated hardware. Images started to be processed in real time such as television standards conversion. (Gregory A. Baxes, 1994).

Now, in 2000s, with developed technologies of fast computers and signal processors digital image processing has become the most common form of image processing due to its versatility and low cost.

## **1.2 Problem statement**

Nowadays, images are used in various fields to help monitoring process. For example, in medication area images are being used to detect symptoms and etc. (John C. Russ, 2002). Therefore, image processing can be used to help performing processing tasks efficiently. However, most image processing software available includes many tasks that sometimes left unused.

The development of this image processing software will help editing process to be done effectively. It requires less space on hard disk; emphasizing only on the crucial image processing functions and the executable program only takes 380KB of size.

## **1.3 Objective**

The objective of this project is to develop a software package to perform basic image processing functions using Visual Basic 6.0.

## **1.4 Scope of Project**

The scopes of this project are;

1. Graphical user interface (GUI) developed using Visual Basic 6.0.
2. Assignment of basic image processing functions including blurring and sharpening, edge detection, colour balancing, white balance and tone (gamma) conversion.
3. Image can be loaded from files and directly from webcam or digicam.
4. Save images into Joint Photographic Experts Group (JPEG) or bitmapped (BMP) images.



## **CHAPTER 2**

### **LITERATURE REVIEW**

#### **2.1 Definition of image processing**

Image processing can be defined as analysis of picture using techniques that can basically identify shades and colours. It deals with images in bitmapped graphic format that have been scanned or captured with digital camera. It also means image improvement, such as refining a picture in a program or software that has been scanned or entered from a video source or in short, image processing is any form of information processing when both the input and output is images.

Image processing is divided into two major branches; image enhancement and image restoration. Image enhancement is to improve the quality of image or emphasize particular aspects within image and to produce image that is different from the original. Whereas image restoration, is to recover the original image after degraded by known effects such as geometric distortion within camera system. Image processing does not reduce the amount of data present but rearranges it. (S. Michael, Lawrence O’Gorman and J.S. Michael, 2006).

## **2.2 Image processing functions/algorithms**

Image processing applications mainly focuses on improving the visual appearance of images to a human viewer and preparing for measurement of the features and structures present. For visual enhancement, the familiarity with the human visual process and an appreciation of what cues the viewer responds to is important. Since many images are processed in the context of reproduction or transmission, printing and display process are included in one of the vital elements. The measurement of images generally requires that features be well defined, either by edges or unique brightness, colour, texture, or some combination of these factors. (S. Milan, H. Vaclac and B. Roger, 1999)

It was also learned that real time imaging systems were used in important application domains, including industrial, medical and national defense. The real time imaging will then be processed in many stages. It will be preprocessed, processed and post processed to produce the image in its desired quality and resolution.

### **2.2.1 Spatial filtering**

An image can be filtered to remove a band of spatial frequencies, such as high frequencies and low frequencies. Where rapid brightness transitions are established, high frequencies will be there. In the other hand, slowly changing brightness transitions represent low frequencies. The highest frequencies normally will be found at the sharp edges or points.

Spatial filtering operations include high pass, low pass and edge detection filters. High pass filters accentuate the high frequency details of image and attenuate the low frequency, creating a sharpening effect. Meanwhile, low pass filters accentuate the low frequency details of image and attenuate the high frequency, creating a blurring effect. Edge detection filters or also known as edge enhancement filters, detect and enhance image edge details.(J. Bernd, 2005).

#### **2.2.1.1 Sharpening**

The main aim in image sharpening is to highlight fine detail in the image, or to enhance detail that has been blurred due to noise or other effects. The Laplacian often used for this purpose. However, image sharpening too can be interpreted in the frequency domain. The higher the frequency, the higher the magnitude of its derivative. (J. Bernd, 2005).

Sharpening emphasizes edges in the image and make them easier to see and recognize. In creating a sharpened image, no new details are actually created. The nature of sharpening is is influenced by the blurring radius used. In addition to that, differences between each pixel and its neighbour too, can influence sharpening effect.

### 2.2.1.2 Blurring

The visual effect of a low pass filter is image blurring. This is because the sharp brightness transitions had been attenuated to small brightness transitions. It have less detail and blurry.

Blurring can be done in spatial domain by pixel averaging in a neighbour. Blurring is aimed to diminish the effects of camera noise, unauthentic pixel values or missing pixel values. For blurring effect, there are two mostly used techniques which are neighbourhood averaging (Gaussian filters) and edge preserving (median filters). A Gaussian blur filter modifies each pixel by looking at its neighbours and computing a "weighted average" of their values, with more weight being given to closer neighbours. The standard Gaussian blur filter found in most image processing programs is isotropic; it blends pixels values equally in all directions. The neighbours to the north and south of a given pixel have just as much weight in the blurring operation as the neighbours to the east and west. For median filtering on the other hand, the outcome is that pixels with outlying values are forced to become more like their neighbours but at the same time edges are preserved. (J. Bernd, 2005).

The blurring effect can improve an image's low frequency details by removing visually disruptive high frequency patterns. By subtracting a low pass filtered image from the original image, a sharpened image can be created. The latter operation is known as unsharp masking enhancement.

### 2.2.1.3 Edge detection

Edges are often used in image analysis for finding region boundaries. They are pixels where brightness changes abruptly. An edge essentially distinguishes between two distinctly different regions or in short, an edge is the border between two different regions. Robert operator, Sobel operator, Prewitt operator, Canny operator and Krish operator are among edge detectors that are often used. Edge detection works by locating sharp changes in the intensity function. A change of the image function can be described by a gradient that points in the direction of the largest growth of the image function.

Edge detection of an image reduces significantly the amount of data and filters out information that may be regarded as less relevant, preserving the important structural properties of an image. There are many methods for edge detection, but most of them can be grouped into two categories, search-based and zero-crossing based. The search-based methods detect edges by looking for maxima and minima in the first derivative of the image, usually local directional maxima of the gradient magnitude. The zero-crossing based methods search for zero crossings in the second derivative of the image in order to find edges, usually the zero-crossings of the Laplacian or the zero-crossings of a non-linear differential expression.

Most edge detection methods work on the assumption that an edge occurs where there is a discontinuity in the intensity function or a very steep intensity gradient in the image. Using this assumption, if the derivative of the intensity values across the image taken and find points where the derivative is a maximum, edge will have to be marked. (J.Bernd, 2005).

### **2.2.2 Diffuse**

Real image data has a level of uncertainty that is manifested in the variability of measures assigned to pixels. This uncertainty is usually interpreted as noise and considered an undesirable component of the image data. (T. Acharya and K. Ray Ajay, 2005).

Image diffusion can smooth away small scale structures thus improving the performances for many image processing algorithms such as image compressions and segmentations.

## **2.3 Introduction to Visual Basic 6.0**

Visual Basic is a high level programming language evolved from the earlier DOS version called BASIC which stands for Beginners' All-purpose Symbolic Instruction Code. (T. Knowlton, K. Barksdale and S. Collings, 2002). Visual Basic is a visual and events driven programming language. These are the main divergence from the old BASIC. In Visual Basic, programming is done in a graphical environment. An important feature in Visual Basic is, its Graphical User Interface (GUI) that presents a pictorial interface to a program. A GUI control is a visual object with which the user interacts via the mouse or keyboard. Users may click on a certain object randomly; therefore each object has to be programmed independently to be able to response to certain actions. In short, Visual Basic programming is made up of many subprograms, each has its own program codes, and each can be executed independently and at the same time can be linked together.

In designing Visual Basic application, firstly the appearance of the window that the user sees will be designed. Then, the events for controls button on the window will be determined. And finally, the event procedures for those events will be written.

When the program runs, Visual Basic monitors the controls in the window to detect any event that a control can recognize (mouse movements, clicks, keystrokes and so on). When Visual Basic detects an event, it examines the program to see if the users have written an event procedure for that event. If event procedure is written, Visual Basic executes the instructions that make up that event procedure and monitors the controls again. These cycles continuously until the application end. Usually, an event must happen before Visual Basic will do anything. Event-driven programs are reactive more than active and that makes them more user-friendly.

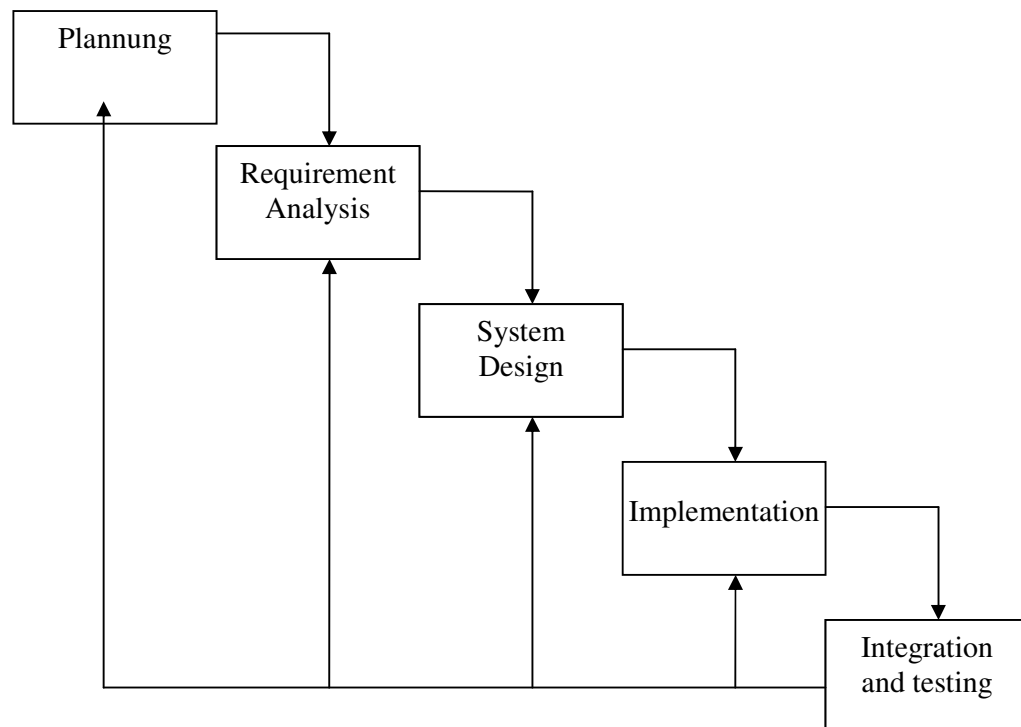
## **2.4 The waterfall model**

In developing a project, system is one of the most important elements to be considered to make sure that the development of the project is smooth and get the expected result. A good system design can described the structure or the flow of the project where by it can be the guideline in managing it. It is also to avoid the project to alter course from the objectives that have been stated or in other words the project follow the guideline based on the objectives

Waterfall Model is considered as the classic approach to the systems development life cycle, and it describes a development method that is linear and sequential. The advantage of using this Waterfall Model is, it allows departmentalization and management control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development

process. Development moves from concept, through design, implementation, testing, installation, troubleshooting and ends up at operation and maintenance. Each phase of development proceeds in strict order without any overlapping or iterative steps. (C.Ghezzi, M. Jazayeri and D. Mandrioli, 1991). Figure 2.1 shows The Waterfall Model discussed.

However, the waterfall model has a defect, which is the limitation for revision. If application is in the testing stage, it is difficult to go back and make changes in the concept stage.



**Figure 2.1** The waterfall model.

The Waterfall Model has five phases; Requirement analysis, System design, Implementation, Integration and testing, and Operation and maintenance. The phases always occur in this order and do not overlap.



The first phase is Requirement analysis. The goal of this phase is to understand the exact requirements of the system and to document it properly. The requirement describes what a system is. This phase produces a large document, known as software requirement specification (SRS) document.

The second phase is System design. The document produced in the preceding phase will be used as a guideline for this phase to be completed. The goal of this phase is to transform the requirement specification into a structure that is suitable for implementation in some programming language. This work is documented and known as software design description (SDD) document. The information contained in the SDD next will be used to begin the coding phase.

The third phase is Implementation. In this phase, design is implemented. If the SDD is complete, the implementation or coding phase proceeds smoothly because there is sufficient information.

The fourth phase is Integration and testing. This is an important phase as the effective during this phase here will contribute to the delivery of higher quality software products, more accurate and reliable results. The purpose of unit testing is to determine each independent module is correctly implemented.

The fifth phase, which is Operation and maintenance will next, maintains the software operation and ensure of its functionality.

## 2.5 Applications of image processing

Image processing technologies are widely used today. L. Rastivlak and P.N. Konstantinus, in 2007 pointed out some of the major applications of image processing technologies and they are in;

- Biological research

Image enhancement – improve visibility of features that are not clear in the original image, using edge detection can be helpful. DNA typing, analysis and matching DNA material can be achieved using image processing.

- Defense/intelligence area

Image processing functions can be used in target acquisition and guidance, recognizing and targeting in real time for smart bombing and missile guidance systems.

- Document processing

Image processing can be used in converting paper documents to a digital image form, compressing the image and storing it for smart archiving.

- Factory automation

Analyzing can be made easier by using visual inspection.

- Medical diagnostic imaging

Computed tomography can be performed where images are being created using multiple image projections. This method is widely used in CT, MRI and PET scanners.

## **CHAPTER 3**

### **METHODOLOGY**

#### **3.1 Implementation of the waterfall model**

As explained in the previous chapter, the Waterfall Model shows the first and foremost step in developing the software is planning. In this stage, the system that will be used will be determined and so does problem statement, objective and scope of the project. Then afterwards, the data will be gathered. The implementations of Waterfall Model are explained in the following subtopics.

##### **3.1.1 Planning**

The software selected to develop this image processing software is Visual Basic 6.0. Interface was made using Visual Basic 6.0 and consists of basic software requirements along with basic image processing functions. The basic image processing functions to be featured mainly consists of sharpening and blurring

(involves high pass and low pass filtering process), grey scale transformation, colours aspect, and edge detection.

Besides the main function of this software, which is image processing; this software too will feature have basic software requirements. User will be able to load files from hard disks, save the project they are working on, print the work, and zoom in and out. Additional functions will include resizing the image, scaling it, and saving it into JPEG or BMP files. Moreover, this software will be able to import images captured from webcam.

### **3.1.2 Requirement Analysis**

Requirement of the image processing software is interfaces made using Visual Basic 6.0. The interfaces then will be inserted with codes that enable interactions between user and the software. Basic image processing functions that will be performed by the software are; sharpening, blurring, diffuse, grey scale, invert, brightness, edge detection, pixelate and colour balance. This software will be able to open files with extensions of .JPG,.BMP and .GIF.

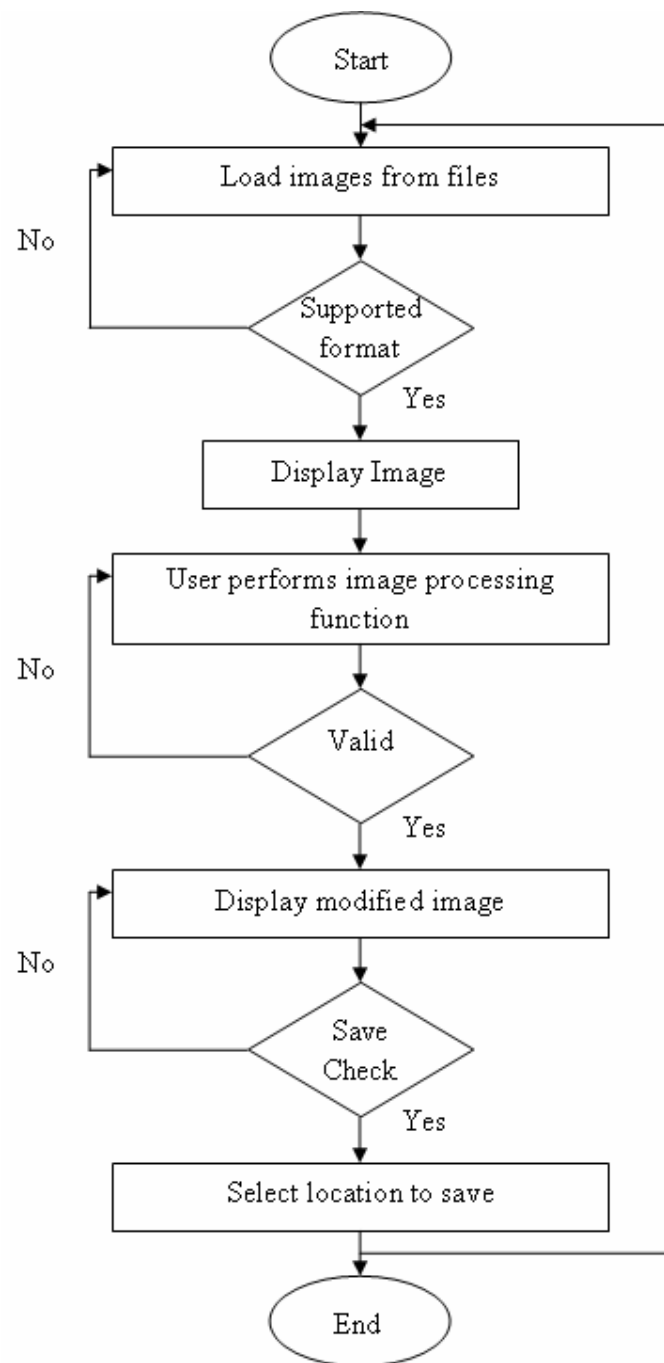
### **3.1.3 System design**

In designing the software, the interface was made first. Then, the interface will be given instructions to perform certain functions required. The flowchart is shown in Figure 3.1. First of all, when the software started user will be directed to the main page. From there, user can load images from files. If the files selected are supported

by the system (in this case GIF, JPEG and BMP), it will be opened and displayed in the main page. If not, the system will not open the file until user selects file with supported format.

The image then, is ready for editing. So from here, user can perform image processing function to the image. If the process is valid, it will be applied to the image and the modified image will be displayed in the main page.

When a user is finished with image editing, he can save the image into specified location before exiting. However, if user tries to exit the program without saving the image, a save check will be performed. This process goes the same if user wants to continue editing with another image.

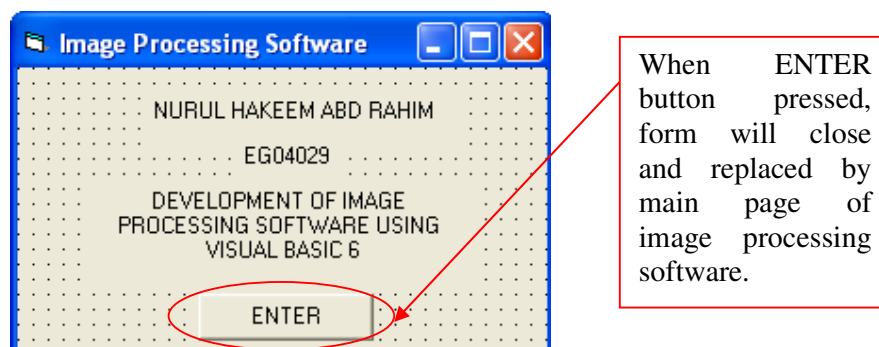


**Figure 3.1** The software operation flowchart

### 3.1.3.1 Software interface

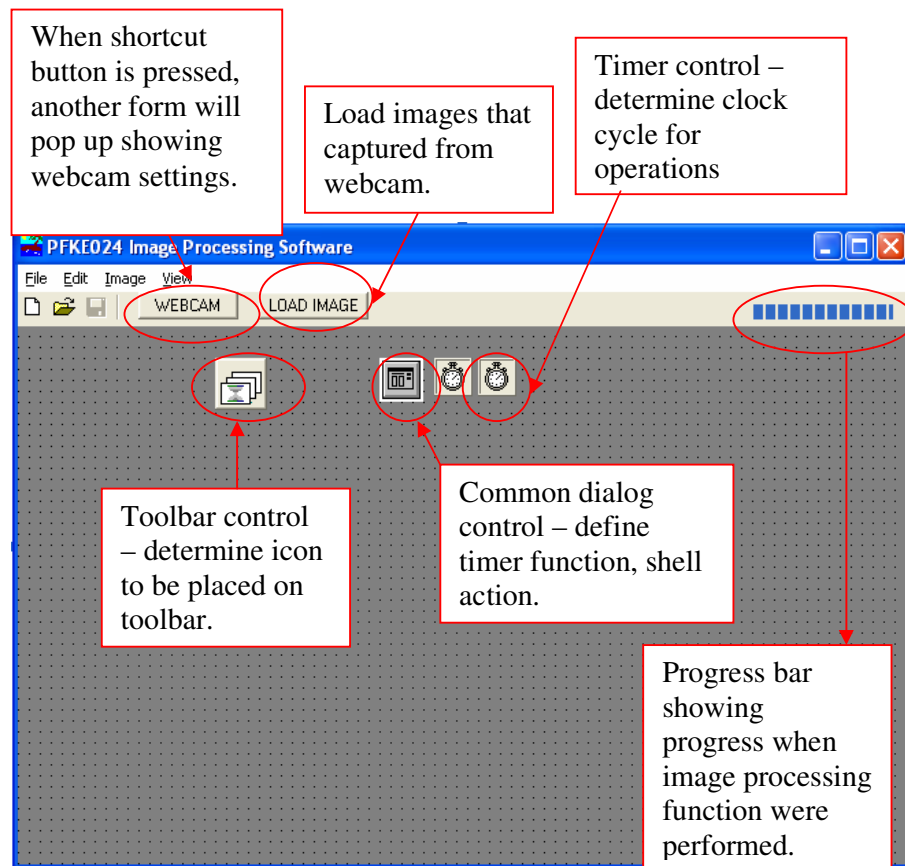
Before inserting the coding and instructions for the software to functioning, the appropriate interfaces were made sure to be completed. However, not all command needs interface. For instance, “Save files”, “Open files” and certain dialog box interfaces need not to be created. This is because the usage of Application Programming Interface (API) in the development of this project. Microsoft Windows consists of libraries of many specialized functions. Some tasks cannot be done by visual basic codes. Therefore, the API calls help to call certain functions from Microsoft Windows’ libraries to complete tasks that could not be completed by simply visual basic codes.

Upon designing the interface, there are forms that needed to be designed. Figure 3.2 shows an introduction form that was made using Visual Basic 6.0. The design of this introduction form consists of a title and a command button. In this project, the title is set to the developer’s name and details. Command button was given a commercial name “ENTER” to make sure user understand the command. The interface after this will go through implementation stage where all the coding and instructions for every action is programmed.



**Figure 3.2** Introduction window.

Then, a design for main page for the software was made. The interface is shown in Figure 3.3. The interface designed for the main page consists of the standard menu toolbar to ensure user understands what to do. Aside from that, a shortcut toolbar was added and it displays shortcuts to some of the functions in the software. In applying the toolbar into the interface, a toolbar control was used to determine the placement of icons on the toolbar. Two command buttons were inserted to specially call the webcam application later. In planning, the “WEBCAM” button was supposed to operate as a shortcut button that will call upon a special webcam window. In the other hand, “LOAD IMAGE” button will be a shortcut to load images from specified folder. The folder that will be called upon is supposed to be the one that stores images from webcam. In designing this interface, timer control were used. Its function is to determine clock cycle for associated operations.

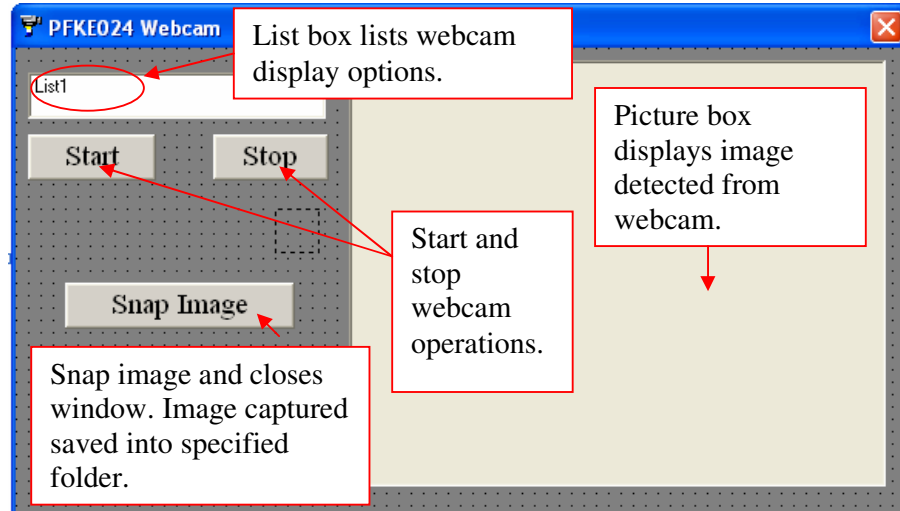


**Figure 3.3** Main page interface.



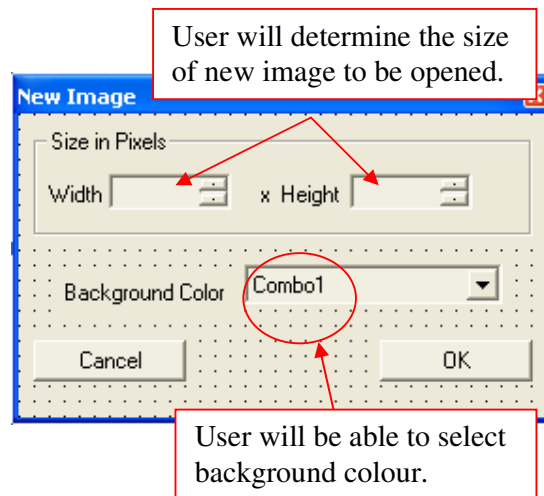
In this main page interface, a progress bar was placed. Later on, it will be programmed to show progress when image processing function were performed.

For designing webcam browser, another form was created. The interface was designed to have a picture box for image displaying purposes as shown in Figure 3.4. A list box was inserted and will be programmed to list webcam display options. There are also three command button that will be set to determine the webcam operation. “START” button will be opening the webcam and displaying the image in the picture box. In the other hand, “STOP” button will terminate the operation and shuts down the picture box. It will be programmed to turn blank after the button clicked. Meanwhile, “SNAP IMAGE” button will be programmed to snap images and save it into specified folder. To avoid multiple image to be snapped at the same time, the button will also be programmed to terminate the whole webcam window and returns to the main page.



**Figure 3.4** Webcam window.

Considering the various sizes of images, a control was designed to enable user to determine the size of the image they will be working on. It will be programmed in the implementation stage where the smallest pixels allowed are 300 x 300 pixels. The largest are 32000 x 32000 pixels. Figure 3.5 shows the interface with a space for user to determine the desired pixels and background color of the image.

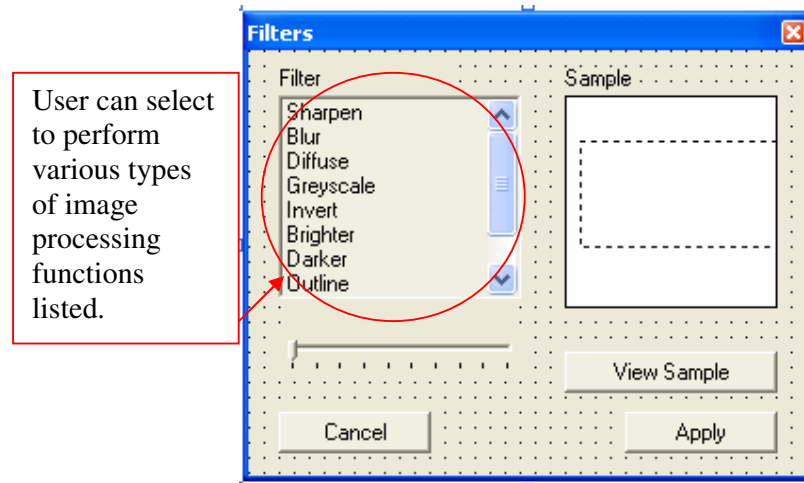


**Figure 3.5** A control to determine image size in pixels.

Another crucial interface to the software is the filter browser. This filter browser will consist of various image processing functions. The options for the process will be put into a window; to make the software appear tidier. The functions are listed and in the implementation stage, the functions listed will be assigned to basic image processing functions. This is shown in Figure 3.6

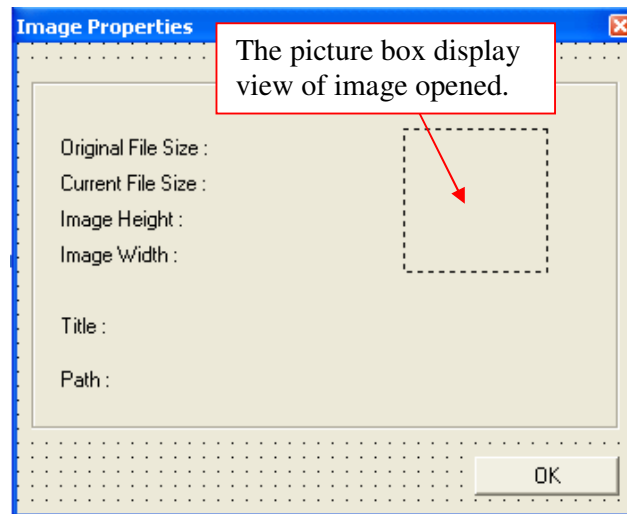
The filter browser enables user to perform basic image processing functions namely sharpen, blur, diffuse, grey scale, invert, brighten, darken, outline, emboss and pixelate. For sharpen, blur and diffuse function, the slider was set for values from 1 to 10. For brighten and darken function, the slider was set for values ranging from 10 to 100. Outline function, emboss and pixelate has the slider to be set for values ranging from 50 to 250, 100 to 200 and 5 to 50 respectively. Meanwhile, for grey scale and invert, both functions do not have slider to be set. The command

button “View Sample” enable user to view the image after selected functions applied. The viewing image appears in the “Sample” picture box.



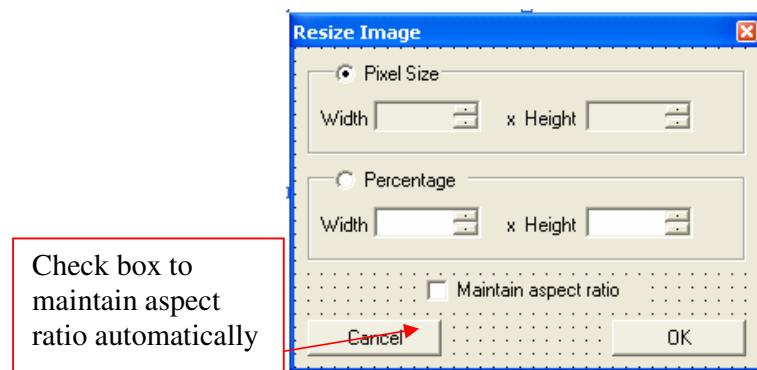
**Figure 3.6** Filter browser interface.

The image properties interface will list the original file size, current file size, image height and width, its title and path. The size of images will be displayed in the form. The image height and width will be displayed in pixels ranging from 300 x 300 up to 32000 x 32000 pixels. More else, the image title and its path in disk too, will be displayed. This is shown in Figure 3.7. There is also a picture box, which will display thumbnail view of image selected



**Figure 3.7** Image properties.

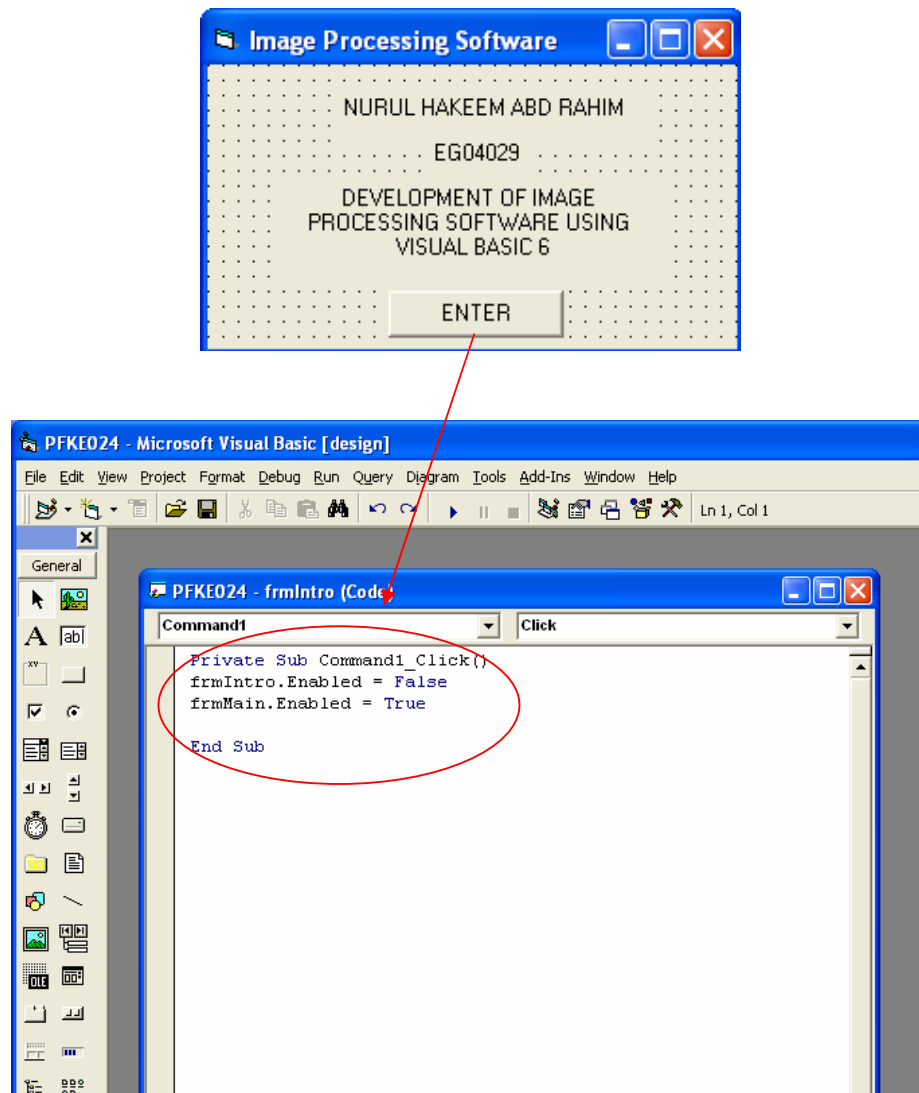
Figure 3.8 shows a resize image interface. In the resize interface, there are two options, whether to resize according to pixel size or percentage. When an image is opened, and resize option is selected, the initial value will be at pixel size, with the size of the image opened as the value. User can set the value to the desired value as long as it does not exceed the setting value for image mentioned before, which is 32000 x 32000 pixels. The check box for aspect ratio will determine whether resizing will automatically follow the aspect ratio or manually set the value of width and height separately.



**Figure 3.8** Resize image interface

### 3.1.4 Implementation

During implementation stage, the interfaces for the software are ready along with the code. In short, in this stage, the software is 90% finished. For instance, the introduction window interface with the coding is shown in Figure 3.9.



**Figure 3.9** Introduction interface.